


# Faire reconnaître votre smartphone, tablette sous Android

- Objet : Faire communiquer un périphérique android avec votre Debian
- Niveau requis :  
[débutant](#)
- Commentaires :
- Débutant, à savoir : [Utiliser GNU/Linux en ligne de commande, tout commence là !](#) 😊
- Suivi :  
[à-tester](#)
- .
- Création par  [PengouinPdt](#) 12/08/2016
- Testé par [trefix](#) le 15/09/2016 : Certains équipement ne sont pas pris reconnu malgré tout 😊 (Voir le sujet de suivi ci-dessous)
- Commentaires sur le forum : [Lien vers le forum concernant ce tuto](#)

## Introduction

Le propos de ce tutoriel est de faire communiquer nos périphériques sous Android (smartphones, et autres tablettes) avec notre station fonctionnant sous Debian (Stable, Sid).



Préférer l'usage des ports USB 2 ... il peut y avoir quelques dysfonctionnements avec les ports USB 3 !

## Périphériques Android testés

- Archos 101 Oxygen : **OK** - *VendorId: 0e79; ProductId: 50f8*
- Cdisplay : **OK** - *VendorId:201e; ProductId: 4289*
- Pipo M8 HD : **OK** - *VendorId: 2207; ProductId: 0010*
- Teclast X89 Kindow : **OK** - *VendorId: 8087; ProductId: 0a5f*
- Xiaomi Redmi : **OK** - *VendorId: 2717; ProductId: ff48*
- Moto E 2015 : **OK** - *VendorId: 18d1; ProductId: 4ee7*
- Samsung S6-EDGE : **OK** - *VendorId: 04e8; ProductId: 6860*

## Vérification

### Vérification côté périphérique

Il faut commencer en effet par paramétrer votre tablette ou smartphone dans le mode de "Débogage USB" nécessaire !

Allumez votre périphérique, puis allez dans le menu "Paramètres", choisissez le menu "Options pour les développeurs", activez-le, puis cocher le choix "Débogage USB".



Si le menu "Options pour les développeurs" n'apparaît pas, c'est qu'il n'est pas actif ... pour l'activer, tapotez une dizaine de fois sur la mention "Numéro de modèle", dans le menu "À propos de la tablette" (ou du téléphone), toujours dans les "paramètres".



Tant que le mode "Débogage USB" n'est pas actif, ou que vous ne pouvez pas l'activer, pas la peine de continuer ... en effet, sans ce mode actif, le périphérique NE PEUT être reconnu !

## Vérification côté ordi

Connectez le périphérique sur le port USB de votre ordinateur :

```
lsusb
```

Pour l'exemple, voici ce que retourne un "Samsung S3", une tablette "Samsung", de modèle "Galaxy Tab" :

```
Bus 006 Device 004: ID 04e8:6860 Samsung Electronics Co., Ltd GT-I9100 Phone [Galaxy S II], GT-I9300 Phone [Galaxy S III], GT-P7500 [Galaxy Tab 10.1]
```

Voici ce que retourne une tablette de marque Pipo M8HD :

```
Bus 002 Device 014: ID 2207:0010
```



L'information à retenir est celle suivant 'ID' ... **La première série de 4 chiffres** est l'identifiant du fabricant, nommé **idVendor** ; **la seconde série de 4 chiffres** est l'identifiant du produit, nommée **idProduct** !

La commande suivante permet de s'assurer que le périphérique Android communique bien toutes ses propres informations à votre station Debian :

```
lsusb -v -d votre_idVendor:
```

Pour l'exemple (dont l'**idVendor est 2717** - remplacez par celle correspondant à votre périphérique) :

```
lsusb -v -d 2717:
```

```
Device: ID 2717:ff40
```

## Device Descriptor:

bLength	18
bDescriptorType	1
bcdUSB	2.00
bDeviceClass	0 (Defined at Interface level)
bDeviceSubClass	0
bDeviceProtocol	0
bMaxPacketSize0	64
idVendor	0x2717
idProduct	0xff40
bcdDevice	2.16
iManufacturer	2 MediaTek
iProduct	3 MT65xx Android Phone
iSerial	4 0123456789ABCDEF
bNumConfigurations	1

## Configuration Descriptor:

bLength	9
bDescriptorType	2
wTotalLength	62
bNumInterfaces	2
bConfigurationValue	1
iConfiguration	0
bmAttributes	0xc0
Self Powered	
MaxPower	500mA

## Interface Descriptor:

bLength	9
bDescriptorType	4
bInterfaceNumber	0
bAlternateSetting	0
bNumEndpoints	3
bInterfaceClass	255 Vendor Specific Class
bInterfaceSubClass	255 Vendor Specific Subclass
bInterfaceProtocol	0
iInterface	5 MTP

## Endpoint Descriptor:

bLength	7
bDescriptorType	5
bEndpointAddress	0x81 EP 1 IN
bmAttributes	2
Transfer Type	Bulk
Synch Type	None
Usage Type	Data
wMaxPacketSize	0x0200 1x 512 bytes
bInterval	0

## Endpoint Descriptor:

bLength	7
bDescriptorType	5
bEndpointAddress	0x01 EP 1 OUT
bmAttributes	2
Transfer Type	Bulk

```

    Synch Type          None
    Usage Type          Data
    wMaxPacketSize      0x0200 1x 512 bytes
    bInterval           0
Endpoint Descriptor:
    bLength              7
    bDescriptorType      5
    bEndpointAddress     0x82 EP 2 IN
    bmAttributes         3
        Transfer Type    Interrupt
        Synch Type       None
        Usage Type       Data
    wMaxPacketSize      0x001c 1x 28 bytes
    bInterval           6
Interface Descriptor:
    bLength              9
    bDescriptorType      4
    bInterfaceNumber     1
    bAlternateSetting    0
    bNumEndpoints        2
    bInterfaceClass      255 Vendor Specific Class
    bInterfaceSubClass   67
    bInterfaceProtocol   1
    iInterface           0
Endpoint Descriptor:
    bLength              7
    bDescriptorType      5
    bEndpointAddress     0x83 EP 3 IN
    bmAttributes         2
        Transfer Type    Bulk
        Synch Type       None
        Usage Type       Data
    wMaxPacketSize      0x0200 1x 512 bytes
    bInterval           0
Endpoint Descriptor:
    bLength              7
    bDescriptorType      5
    bEndpointAddress     0x02 EP 2 OUT
    bmAttributes         2
        Transfer Type    Bulk
        Synch Type       None
        Usage Type       Data
    wMaxPacketSize      0x0200 1x 512 bytes
    bInterval           0
Device Qualifier (for other device speed):
    bLength              10
    bDescriptorType      6
    bcdUSB               2.00
    bDeviceClass          0 (Defined at Interface level)
    bDeviceSubClass       0
```

```
bDeviceProtocol      0
bMaxPacketSize0      64
bNumConfigurations    1
Device Status:        0x0001
Self Powered
```

Un autre moyen de confirmer la prise en charge du périphérique Android, juste après l'avoir connecté au port USB, est de taper la commande suivante :

```
dmesg | grep -i "usb .* idVendor" | tail -1 | sed 's/\[ /\[/g'
```

```
[279059.769436] usb 1-2: New USB device found, idVendor=2717, idProduct=ff40
```

ou pour ceux qui ont **systemd** :

```
journalctl -k | grep -i "usb .* idVendor" | tail -1 | sed 's/\[ /\[/g'
```

```
août 14 10:42:57 ptb-zou kernel: usb 1-2: New USB device found,
idVendor=2717, idProduct=ff40
```



Ce qui est nécessaire à retenir comme informations sont l'identifiant du fabricant "idVendor", et l'identifiant du périphérique nommé "idProduct" ... elles nous serviront dans la phase de configuration !

Dans le cas de ce périphérique connecté, c'est **idVendor=2717** et **idProduct=ff40**

## Installation

Commençons par l'installation des binaires nécessaires, dans votre ordinateur Debian :

```
apt install android-tools-adb
```

## Configuration

### Modifications de fichiers

Deux fichiers à modifier/créer sont nécessaires, dans Debian :

1/ le premier est un fichier système udev qui s'occupe de la gestion usb :

```
nano /etc/udev/rules.d/51-android.rules
```



Il faut modifier la valeur de la variable ATTR{idVendor} par l'identifiant du fabricant trouvé dans le contexte précédent de la vérification !

Pour continuer notre exemple avec le dernier appareil testé (idVendor=2717 et idProduct=ff40) il faudrait donc remplir ce fichier ainsi:

```
SUBSYSTEM=="usb", ATTR{idVendor}=="2717", MODE="0666", GROUP="plugdev"
```

2/ le deuxième fichier qui est intéressant à créer ou à modifier s'il existe déjà, est le fichier personnel suivant :

```
nano ~/.android/adb_usb.ini
```

Il faut y ajouter l'information de l'identifiant du fabricant, un '0x' suivi de idVendor.

Toujours avec notre exemple, il faudrait donc écrire :

```
0x2717
```



Si vous avez plusieurs périphériques Android, écrivez chacune des informations relatives à ces équipements ligne par ligne ... dans chacun de ces deux fichiers !

## Gestion utilisateur

Vérifier que votre utilisateur linux soit bien dans le groupe "plugdev" !

```
getent group plugdev
```

Vous devez avoir pour retour l'équivalent de ceci :

```
plugdev:x:46:votre_user_name
```

Si ce n'est pas le cas, que votre identifiant linux n'apparaisse pas, il va nous falloir l'ajouter au groupe en question :

```
usermod -G plugdev votre_user_name
```

## Redémarrer le système udev

Maintenant que les fichiers sont modifiés, il est nécessaire de redémarrer le sous-système udev pour permettre la gestion correcte du périphérique :

```
service udev restart
```

## Utilisation

Maintenant que votre périphérique Android est non seulement connecté en USB, reconnu puis correctement pris en charge par votre station Debian, il suffit d'utiliser l'outil adb ainsi :

```
adb devices
```

Pour l'exemple, vous devriez avoir le retour d'information suivant :

```
List of devices attached  
WAWJM5UMUW device
```



Pour obtenir l'aide relative à l'outil adb, il suffit de taper dans votre terminal, la commande portant son nom : adb ... qui vous restituera les informations en anglais

## Copier fichiers vers/depuis

### Copier vers le périphérique

```
adb push -p /repertoire_local_machine/fichier  
/repertoire_peripherique_android
```

### Copier depuis le périphérique

```
adb pull -a -p /repertoire_peripherique_android/fichier  
/repertoire_local_machine
```

L'option -a copie les informations telle que le temps, et le mode du fichier.



Si le répertoire local de la machine n'est pas spécifié, le fichier depuis android sera copié dans le répertoire d'où est exécutée la commande adb.

## Sauvegarde

La commande principale est :

```
adb backup -f nom_fichier_sauvegarde.ab
```



Si l'option -f n'est pas spécifiée, le fichier backup.ab sera créé dans le répertoire courant où est exécutée la commande de sauvegarde...

## Sauvegarder tout

```
adb backup -f nom_fichier_sauvegarde.ab -all
```

Cette commande implique la sauvegarde de toutes les applications installées, dont les applications systèmes.



Cette option intègre l'option `-system` - pas besoin de la spécifier !

## Sauvegarde juste les applications

```
adb backup -f nom_fichier_sauvegarde.ab -apk -obb
```

Cette commande sauvegarde les fichiers apk des applications installées.



- L'option par défaut est `-noapk` qui signifie que les fichiers apk ne seront pas sauvegardés !
- L'option `-obb` implique la sauvegarde des fichiers relatifs aux applications installées, tels que fichiers de sauvegarde, de config, etc ... - par défaut, c'est l'option `-noobb` qui est active !

```
adb backup -f nom_fichier_sauvegarde.ab package1 package2 package_n
```

Sauvegarde juste les applications dont les noms sont concernés !

## Sauvegarde SD Carte



**ATTENTION : Il est bien sûr nécessaire d'avoir une SD Carte dans votre appareil ...**

```
adb backup -f nom_fichier_sauvegarde.ab -shared
```

Sauvegarde le contenu de la SD Carte, ainsi que de tout répertoire de stockage partagé.



Par défaut, c'est l'option `-noshared` qui est active et implique la non sauvegarde !



## Sauvegarde Système

```
adb backup -f nom_fichier_sauvegarde.ab -system
```



Par défaut, c'est l'option `-system` qui est activée, ce qui a pour effet d'inclure les applications systèmes !

## Sauvegarde Pertinente

Le moyen pertinent de sauvegarder tout correctement, sans s'occuper du contenu de la SD Carte, est :

```
adb backup -f nom_fichier_sauvegarde.ab -apk -obb -all
```

## Sauvegarde des partitions

L'outil adb peut servir à sauvegarder indirectement les partitions de votre tablette ...

Pour cela, il faut télécharger l'outil [rkdump](#) !

Puis l'installer ainsi :

```
adb push rkdump /data/
```

```
adb shell chmod 0755 /data/rkdump
```

Pour l'utiliser l'outil rkdump, veuillez lire son [tutoriel](#) ...

## Restauration

```
adb restore nom_fichier_sauvegarde.ab
```



**Attention : Tout ce qui concerne le contenu du fichier de sauvegarde sera restauré !**

## Erreurs connues

### error while loading shared libraries: libncurses.so.5

Si vous fonctionnez avec une distribution GNU/Linux 64 bits, c'est qu'il vous manque les librairies pour

l'architecture 32 bits !

### 'adb devices' : rien n'est affiché

Si rien n'est reconnu, et donc qu'aucun device n'apparaît, c'est que vous avez peut-être mal renseigné le fichier `/etc/udev/rules.d/51-android.rules` et/ou n'avez pas redémarré le système udev !

Si malgré tout, vous êtes absolument sûr d'avoir bien rempli le fichier en question, et d'avoir redémarré udev, il peut être intéressant d'effectuer la manipulation suivante, vérifiez aussi que vous avez configuré le fichier personnel `~/.android/adb_usb.ini`

Si c'est le cas, redémarrer le serveur adb :

```
adb kill-server
```

```
adb devices
```

### 'adb devices' : offline

Lorsque je tape `adb devices`, le périphérique est reconnu mais affiché offline.

```
adb devices
```

```
List of devices attached
WAWJM5UMUW      offline
```

Il vous faudra alors :

1. désactivez l'option "Débogage USB"...
2. déconnectez/reconnectez votre tablette à l'ordinateur, en débranchant/rebranchant le câble USB.
3. activez à nouveau l'option "Débogage USB"

À ce moment-là, essayez à nouveau la commande `adb devices` ; vous devriez voir apparaître votre périphérique ... correctement !



Après l'activation "débogage usb", il est possible que le smartphone vous demande d'accepter le pc sur lequel est connecté le smartphone... validez, cela fonctionnera :)

### 'adb devices' : unauthorized

Lorsque je tape `adb devices`, le périphérique est reconnu mais affiché **unauthorized**.

```
adb devices
```

```
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
List of devices attached
CQ7SFQMR95LBHQVW      unauthorized
```

Regardez votre téléphone, une fenêtre vous demande, normalement, d'autoriser l'association entre le téléphone et votre ordinateur. Accordez !

Puis relancez la commande `adb devices`.

**'adb devices' : ???????????? no permissions**

Lorsque je tape `adb devices`, le périphérique n'est pas reconnu et il est affiché **????????????? no permissions**.

```
adb devices
```

```
List of devices attached
?????????????      no permissions
```

Il vous faut tuer le serveur, le redémarrer avec des droits administrateur ensuite le périphérique apparaîtra !

```
adb kill-server
```

```
adb start-server
```

```
daemon not running. starting it now on port 5037
daemon started successfully
```

```
adb devices
```

```
List of devices attached
WAWJM5UMUW device
```

---

## Autres documentations

Voilà ce tutoriel est terminé ... il ne traite pas de l'usage plus complet des outils 'adb', 'fastboot' ...

Au besoin, peut-être pourriez vous lire ces autres tutoriels :

- [dump-android](#).
- pour KDE : [kdeconnect](#), [connecter-vos-peripheriques-mtp](#) ...

From:  
<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:  
<http://debian-facile.org/doc:materiel:reconnaitre-peripherique-android>



Last update: **10/03/2018 14:02**